HyperLogLog: Analysis and implementation of an improved algorithm

Dequeker Chloé, Ziat Ghiles

February 2015

Contents

1 Introduction

2 cardinality estimation problem

Finding the number of distinct elements in a data set with duplicates is a well-known problem which is applies in many fields.

The naive solution to this problem is to examine for each element of the data stream its belonging to a data structure $\mathcal{D}$. If $\mathcal{D}$ doesn't contain the element we add it to the data structure. At the end of the process, the cardinality of the data stream is equal to the size of $\mathcal{D}$.

This solution gives the exact answer but it easy to see that is scales very badly as the size of the data stream grows.

3 HyperLogLog

4 HyperLogLog++

4.1 transition to 64 bits

4.2 Bias estimation and correction

4.3 Memory optimization

4.3.1 Sparse representation

4.3.2 Dense representation

4.3.3 Compressing the sparse representation

Since the temporary set used in the sparse representation is merged with the list before it gets too large, it's not very relevant to perform a compression on it. However, having no such a size limit for the sorted list, we'll try to reduce its memory usage playing on two points:

- Using fixed-size integers as it is common practice in many langages may here result in a waste of memory space.
- Since the manipulated list is sorted, we can take advantage of this information.

varint encoding

5 Conclusion